PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.:	09/902,184	Confirmation No.: 7537
Applicant(s):	Robert Craig Murphy	
Filed:	July 10, 2001	
Art Unit:	2172	
Examiner:	A. Ly	
Title:	SYSTEMS AND METHOD FOR INTEGRATING ELECTRONIC STORAGE FACILITIES	

Docket No.: 023895/257911
Customer No.: 00826

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

DECLARATION UNDER 37 C.F.R. § 1.131

Sir:

We, Robert Craig Murphy, Karen Carter, Ceryl Medua, Rhadee Resma, Richard Sharp, Brian Wong, and Claudia Woodruff, hereby declare and state that:

1. We are the inventors of the claimed invention of the above-identified U.S. Patent Application Serial No. 09/902,184.

2. We have read and understand U.S. Patent Application Publication No. 2003/0046309 to McGrath et al. ("McGrath"), which was filed June 13, 2001 and published March 6, 2003, and which was relied upon by the Examiner in the Official Action mailed January 25, 2007 as disclosing or suggesting portions of independent Claims 1, 7, 13, 19, 23, and 25 of the above-referenced application. This Declaration is filed to establish actual reduction to practice prior to the filing date of McGrath.

3. Prior to June 13, 2001, the filing date of McGrath, we actually reduced embodiments of the claimed invention to practice. In particular, we developed a prototype of the claimed invention that worked for its intended purpose, as described below, thereby reducing to practice embodiments of our invention as described and claimed in the subject application, which is generally directed to methods, a computer, and systems for sharing customer information among a plurality of electronic storage facilities. In support of this statement, we have attached Exhibits 1-5. Although the dates of Exhibits 1-5 are not shown, these exhibits are dated prior to June 13, 2001 (*See MPEP § 715.07: Establishment of Dates*).

4. In support of the foregoing statement regarding actual reduction to practice, we hereby submit the best available copy of the following documents:

a. Exhibit 1 – Customer DNA: document disclosing the Customer DNA functionality.

b. Exhibit 2 – Customer DNA Phase I, Detail Design: document disclosing the Customer DNA functionality and program specifications for implementing such functionality.

c. Exhibit 3 – Customer DNA Phase I, Interface Specifications: document disclosing the protocols, data flows, processes, and procedures between the Customer DNA system and external applications.

d. Exhibit 4 – CRM at the Travel Agency desktop: presentation illustrating a CRM system including the Customer DNA functionality.

e. Exhibit 5 – Status Report disclosing that a prototype of the CRM system including the Customer DNA functionality had been developed.

5. Exhibits 1-5 provide support that we reduced to practice the methods, computer, and systems of embodiments of the claimed invention for sharing customer information among a plurality of electronic storage facilities. Generally, Exhibits 1-5 illustrate the Customer DNA ("CDNA") functionality, which includes creating a unique customer identifier for providing a consolidated view of the customer among various databases.

6. More specifically, Exhibits 1-5 disclose the methods, computer, and systems of embodiments of at least independent Claims 1, 7, 13, 19, 23, and 25 of the present application. In this regard, Exhibits 1-3 disclose the CDNA functionality, which generally includes creating a customer identifier for sharing customer information among electronic storage facilities. Namely, Exhibits 1-3 disclose that the CDNA system includes a mass data store (i.e., Master DNA index) including a first data record (e.g., index record 1) identifying information for a customer having an associated first customer identifier (e.g., key information 1). Exhibits 1-3 also disclose that information identifying the customer may be received from an electronic storage facility (e.g., PNR index or RMS) that includes a second customer identifier (e.g., key information 2) that is stored in a second data record (e.g., index record 2). Exhibits 1-3 also disclose that a determination is made whether the identifying information in the first and second data records are associated with the same customer. Furthermore, Exhibits 1-3 disclose that an identifier (i.e., DNA number) is assigned based on the determination and cross referenced with the identifying information in the first and second data records and that the identifying information may be provided to a requesting electronic storage facility using the assigned identifier. Exhibits 1-3 further disclose that the CDNA system may receive identifying information for the customer from a subsequent electronic storage facility, store the identifying information in a list of electronic storage facilities, and cross reference the identifying information with the previously assigned identifier.

7. Exhibits 2-5 disclose that embodiments of the claimed invention were reduced to practice. Namely, Exhibits 2 and 3 disclose detail design and interface specifications required to develop software for implementing the CDNA functionality. Exhibit 4 illustrates a Customer Relationship Management ("CRM") system that was generally employed to use customer information (e.g., profile, interests, previous travel, etc.) for various functions, such as targeted marketing. Exhibit 4 also illustrates screen shots of the operational CRM system. The CRM system included the CDNA functionality of the claimed invention. In particular, the "Compaq ZLE" component provided the CDNA system, which included the functionality for cross-

referencing customer information from the Relationship Management System ("RMS"), Buyer Offload Analysis Data Distributor Passenger Name Record ("BOA PNR"), and Buyer Offload Analysis Data Distributor ("BOA DD") with an assigned, unique identifier for each customer that was stored within a mass data store. Moreover, Exhibit 5 specifically states that a prototype of the CRM system was constructed and completed (e.g., see "Milestones" on p. 19).

8. All of the work we did in connection with this invention was carried out in the United States.

9. We hereby declare that all statements made herein of our own knowledge are true, and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Robert Craig Murphy

Robert Craig Murphy

Karen Carter

Ceryl Medua

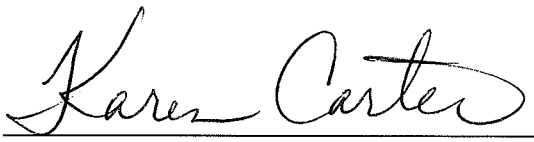
Rhadee Resma

referencing customer information from the Relationship Management System ("RMS"), Buyer Offload Analysis Data Distributor Passenger Name Record ("BOA PNR"), and Buyer Offload Analysis Data Distributor ("BOA DD") with an assigned, unique identifier for each customer that was stored within a mass data store. Moreover, Exhibit 5 specifically states that a prototype of the CRM system was constructed and completed (e.g., see "Milestones" on p. 19).

8. All of the work we did in connection with this invention was carried out in the United States.

9. We hereby declare that all statements made herein of our own knowledge are true, and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Robert Craig Murphy



Karen Carter

Ceryl Medua

Rhadee Resma

referencing customer information from the Relationship Management System ("RMS"), Buyer Offload Analysis Data Distributor Passenger Name Record ("BOA PNR"), and Buyer Offload Analysis Data Distributor ("BOA DD") with an assigned, unique identifier for each customer that was stored within a mass data store. Moreover, Exhibit 5 specifically states that a prototype of the CRM system was constructed and completed (e.g., see "Milestones" on p. 19).

8. All of the work we did in connection with this invention was carried out in the United States.

9. We hereby declare that all statements made herein of our own knowledge are true, and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Robert Craig Murphy

Karen Carter

Ceryl Medua

Rhadee Resma

referencing customer information from the Relationship Management System ("RMS"), Buyer Offload Analysis Data Distributor Passenger Name Record ("BOA PNR"), and Buyer Offload Analysis Data Distributor ("BOA DD") with an assigned, unique identifier for each customer that was stored within a mass data store. Moreover, Exhibit 5 specifically states that a prototype of the CRM system was constructed and completed (e.g., see "Milestones" on p. 19).

8. All of the work we did in connection with this invention was carried out in the United States.

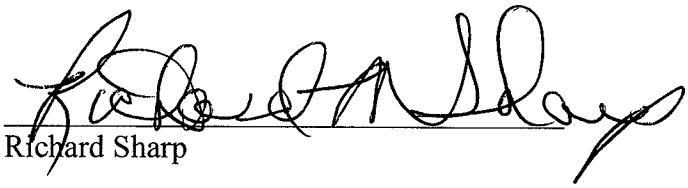
9. We hereby declare that all statements made herein of our own knowledge are true, and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Robert Craig Murphy

Karen Carter

Ceryl Medua

Rhadee Resma

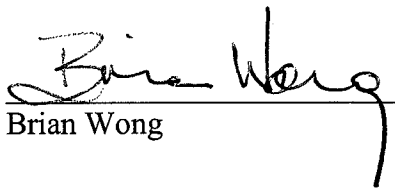


Richard Sharp

Brian Wong

Claudia Woodruff

Richard Sharp



Brian Wong

Claudia Woodruff

Richard Sharp

Brian Wong

Claudia Woodruff

CUSTOMER DNA

EXHIBIT 1

Table of Contents

About This Document

Date Created:

Date Last Revised:

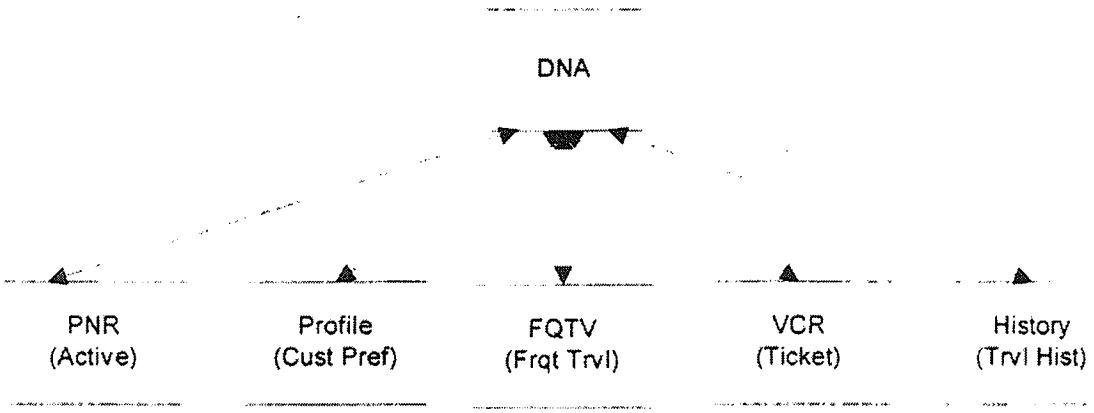
Authored By: Rhadee Resma/Brian Wong

Revised By:

Reviewed By:

Customer DNA Concept from Position Paper

"Customer DNA is a series of functions and/or systems that will enable Sabre to access multiple sources of data about a customer. The DNA will provide a mechanism for the **TPF system** to obtain pertinent information as quickly as possible about the customer such as; other active PNR records in the system, profile information containing customer preferences, frequent traveler program information, ticket information, and historical travel information."

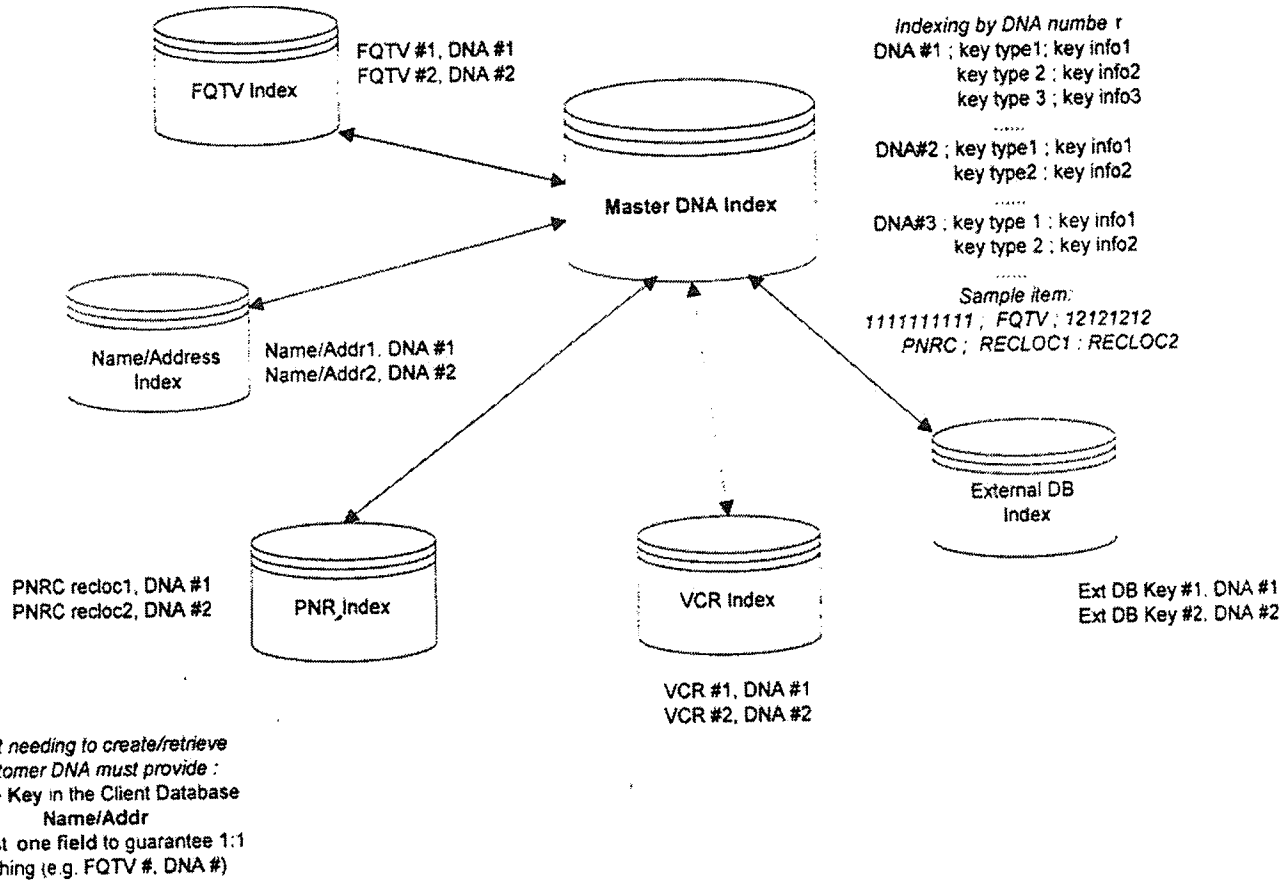


Our concept of Customer DNA

- **Requirements:**

1) Create one central repository of customer key data to be able to access the various data stores in Sabre.
2) Given a prime key element from one system, enable the retrieval of the Customer DNA number and the associated prime keys to allow access to other systems.
3) With minimal application changes (particularly from external applications), enable the access of all the databases linked by Customer DNA.
4) Provide a source of obtaining a customer-centric view of the traveler.
5) Design an architecture that will allow data synchronization.

- **Our solution:**



- ***Database component : Master DNA Index*** - Create a Master DNA Index which is indexed by DNA number and will have the key information to access the various data stores. The following fields will be present in the Master DNA index:
  DNA Number ; Key Type ; Key Information (where Key Type and Key Information are repeating)
  For example:

DNA # 121221312312 ;	FQTV ;	AA 1234454
	PNRC ;	QJRAUS ; VXYZJS ; RHJISQ
	PDW ;	QVQSDF
	NAME/ADDR ;	JONES/A MR ; 123 MAIN ST.

- ***Database component : Individual Database Index*** - For each type of retrieval method, an index record will be created. For example, retrieval by FQTV number will require a FQTV index (index item = FQTV # and DNA number). Retrieval by PNRC record locator will require a PNR index (index item = REC LOC and DNA number).

- ***Database component : Name/Address Index*** – A name/address index is necessary not for retrieval purposes (simply because positive customer identification cannot be guaranteed from this index), but for facilitating the nightly file maintenance process required to eliminate dupe items.

- ***Input Requirement*** - Client requiring a new DNA number must supply its **database key information** (so we can index their data into the individual database index and the master DNA index). In addition, the **name and address** must be provided as supplemental information to be able to index the item into the name/address index. For positive customer identification, at least **one additional prime key information** must be provided (e.g., FQTV number, DNA number).

- ***Generation of New Customer Number :Algorithm vs. Sequential Assignment*** - Customer DNA may be assigned sequentially starting from 1. It may also be derived using an algorithm that will use input fields such as traveler name, address and/or phone number. A **test database** will be created to ensure that the algorithm chosen will result in an even spread of database items.

- ***Platform*** - The DNA indices may be in TPF or UNIX. It may be a TPFDF file located in the PNRC Complex or it may be a Relational Database in UNIX. Due to the volume of travelers and the message rate of the potential applications that will utilize Customer DNA (e.g., End Transaction and sell/availability), we recommend TPF. It is the safest approach, is proven and guaranteed to work, and has the speed and capacity required to handle massive amount of data and traffic. A UNIX solution may be explored after initial implementation if necessary.

- ***Services*** - The following services will be written to support Sabre's service-oriented architecture:
    1) Generate Customer DNA
    2) Retrieve Customer DNA
    3) Delete a Customer DNA item
    4) Update key information
    5) Get Customer Details/Experience * (*Service calling another service*)

- ***Change Management*** – If the prime key changed, the old and new key information must be provided in order to retrieve the original DNA number and assign it to the new key and to de-index the old key. If the traveler address changed, the indexing of the new address and de-indexing of the old address must occur. In addition, **a publish and subscribe approach** can be implemented to allow synchronization of the traveler address across the various databases. This may consist of the DNA system publishing the address change via **MQ series** and the relevant systems subscribing to the change and de-queuing the specified queues.

- ***Handling of Duplicate Customer DNA*** – If the traveler does not have a unique identification such as FQTV number and will come into the DNA system identified only by its name and address, positive customer identification is impossible. For this type of passenger, a new Customer DNA will have to be generated every time. To handle the dupe items that will result, a nightly file maintenance job will be needed for database clean up. This utility will have to identify all the dupe suspects and eliminate the dupes. It will have to access databases such as RMS, AATMS and PNRC to be able to analyze the dupe suspects. If in the future the database size becomes too big to easily manage, it may be necessary to have an offline copy of the Customer DNA system (i.e., a DSS in UNIX) where the dupes checker can be run.

- ***Recoup*** - Client applications that have recoup facility will have to enhance their process to send the deleted items to the Customer DNA system. An item purged/deleted from a client database must also be deleted from the DNA indices. If the last DNA item is being deleted, the entire DNA information including the DNA number must be deleted. This means the DNA system (TPF) must subscribe to the recoup event published by the individual databases. If the volume of updates due to recoup proves to be too high, it may be necessary to have an offline copy of the Customer DNA system for decision support usage (i.e. a DSS in UNIX). However, recoup from PNRC should not be a problem because the DNA system will be in the same complex.

- ***Viewership of Customer Details*** - Each application will have to enforce its own viewership/security rules. This will be made possible by the client sending Point-of-Sale information to the server which will use it to determine if viewership is allowed or not.

- ***Viewership of Customer DNA Number -*** A feature will be implemented within the Customer DNA system to ensure that the Customer DNA number is sent only to internal clients within Sabre and not to external clients. The originating system information will be checked to ensure that the DNA number is not published outside of Sabre and that only an acknowledgement is sent.

- ***Backup/Restore of DNA Indices -*** A backup/restore facility will be implemented to ensure that data recovery of index information is possible in case of data corruption.

- ***Phase 1 - RMS, PNR Data Warehouse and SAM (Sales and Alliance Manager)***
  These systems must be able to "subscribe" to any changes in the DNA indices. For example, if the address changed, each client database must be updated to reflect the change. They must also have **expiration subscription** to ensure that duplicate items expired in the DNA indices are also expired in their own respective database.

  - ***Relationship Management System (RMS) – Customer Profile:***

- When a profile is created, a new Customer DNA number will be required. RMS will go to the Customer DNA system with the following input - name, address, phone, FQTV number if present, User ID if present.

- The **DNA number** generated will be indexed into the Master DNA Index and the Name/Address index. It will also be returned to RMS where it will be used as an **index**. This will allow other applications to retrieve customer details using the DNA as index.

- **OLTP application** in RMS will allow query of customer details using the Customer DNA as key.

- When customer name or phone number changes, RMS must retrieve the Master DNA index to update key information. Old name/phone must be provided along with the new name/phone. This is to index the original DNA number into the new location (derived from the new data).

  - ***PNR Data Warehouse (PDW) – Travel History:***

- When a new PNR is ended and the traveler does not have a DNA number, a new one will be generated. This will have to happen in **End Transaction**, prior to the filing down of the PNR.

- The Customer DNA number will be **stored in the PNR**. The field will be name associated.

- DNA number will be included as PNR data when logging to the BOA Data Distributor.

- After logging the PNR to BOA, an interface to the Customer DNA system will be necessary to index the PNR that had been logged to BOA.

- **OLTP application** will be needed to allow query using the BOA record locator as key.

➢ ***Sales and Alliance Manager (SAM) – Target Marketing at Sell and Availability:***

- The SAM DSS will access other databases like RMS to get customer data using DNA number and FQTV number.

- The Sell and Availability packages will be changed to send the DNA number to the SAM runtime database to get targeted marketing messages. Currently, only the FQTV number and traveler name are sent.

➢ ***Phase 2 - PNRC, AATMS, VCR -*** These systems must be able to "subscribe" to any changes in the DNA indices. For example, if the address changed, each client database must be updated to reflect the change. They must also have **expiration subscription** to ensure that duplicate items expired in the DNA indices are also expired in their own respective database.

➢ ***PNRC – Active PNRs:***

- When a new PNR is created, or when new names are added to an existing PNR, an interface with the DNA system will be necessary to index the new names into the DNA indices.
- The Nightly File Maintenance process for the PNRs must be modified to reflect the PNR purging in the DNA indices.

➢ ***AATMS – Frequent Traveler data:***

- When a new Aadvantage record is created, an interface with the DNA system will be necessary to index the new member into the DNA indices.

- **OLTP application** in AATMS will allow query using FQTV as key.

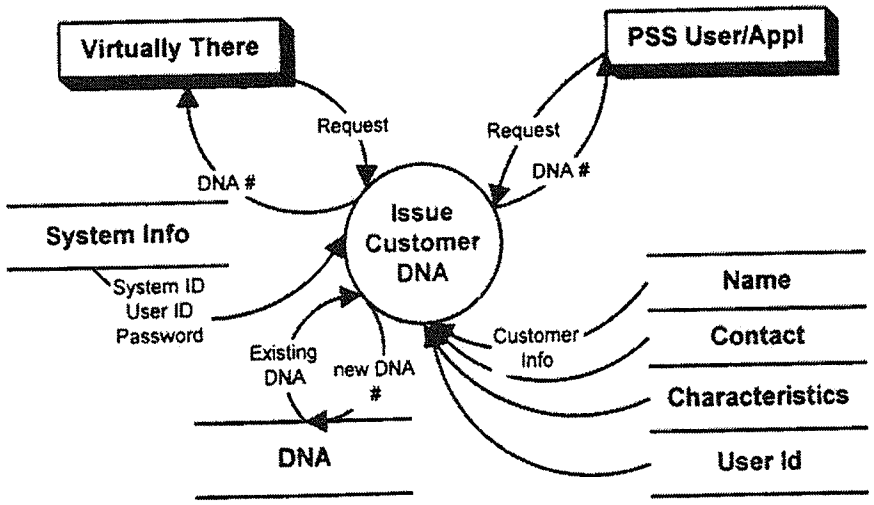
➢ ***VCR – Ticket Information:***

- When a new VCR item is created, indexing into the DNA indices must occur.

- A **VCR query utility** will be created to allow query of all VCRs for a particular traveler.

Process Diagrams

1. Issue Customer DNA



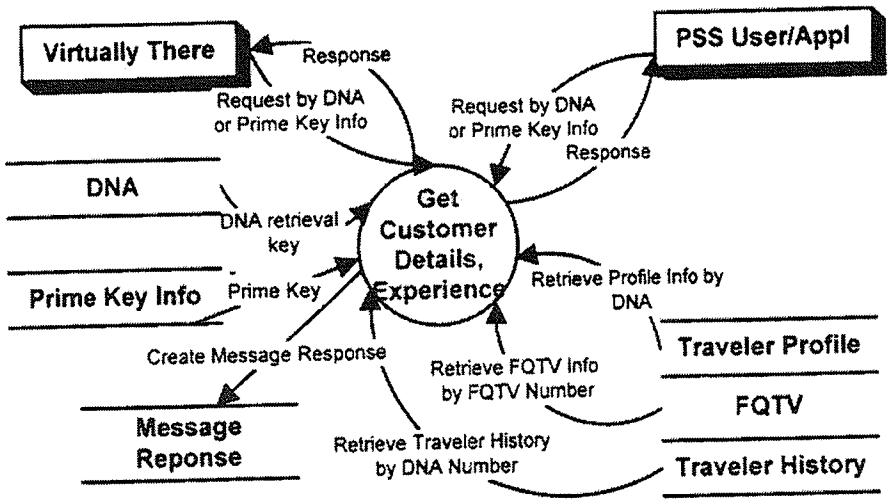
Process Narrative:

Query Customer DNA; if exists indicate it already exists
Calculate DNA number
Record unique id plus data needed to calculate key: calculated DNA number, name, phone, address.
Return the DNA number

Additional Comments:

We record the DNA information into the Customer Master Index.

2. Get Customer Details/Experience



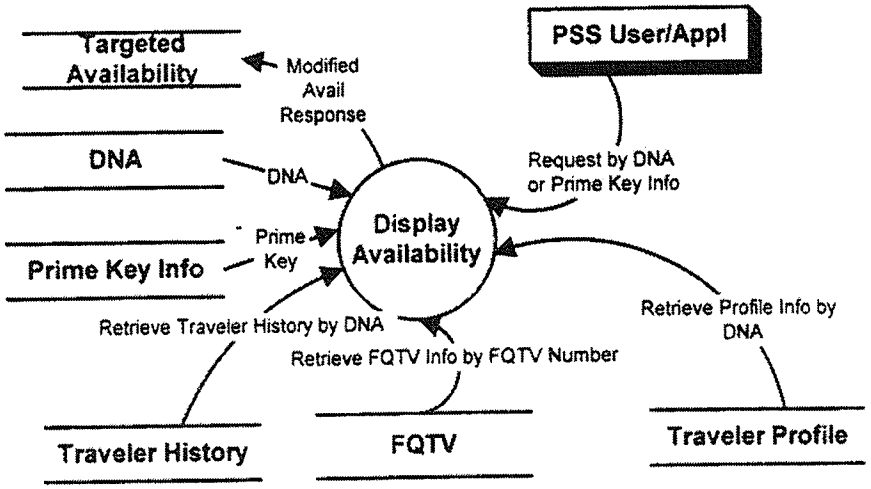
Process Narrative:

Requestor Requests Customer Details by Customer key information
Requestor Requests Customer Experience by Customer key information
Send Details/Experience Response to Requestor

Additional Comments:

A request will be made to retrieve data from each system that carries customer information. (e.g. one for RMS and one for PNR data warehouse).
It is the application that is responsible for accessing each respective database.

3. Display Availability



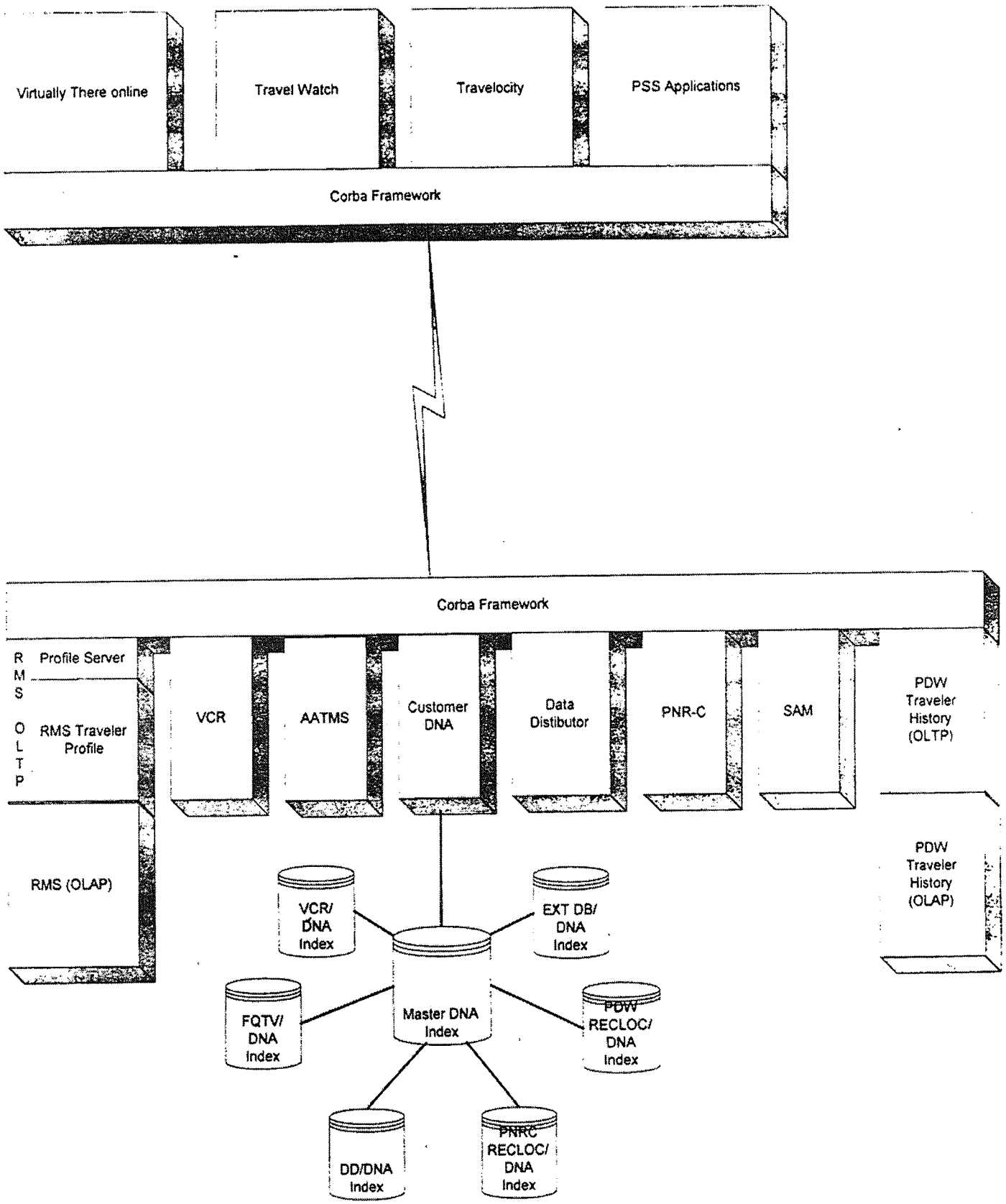
Process Narrative:

Requestor Requests Availability (Customer DNA or Prime Key Information is available)
Retrieve Customer Details by Customer key information
Retrieve Customer History by Customer key information
Retrieve Customer Frequent Traveler Data by Customer key information
Targeted Availability Response is assembled
Targeted Availability Response is sent

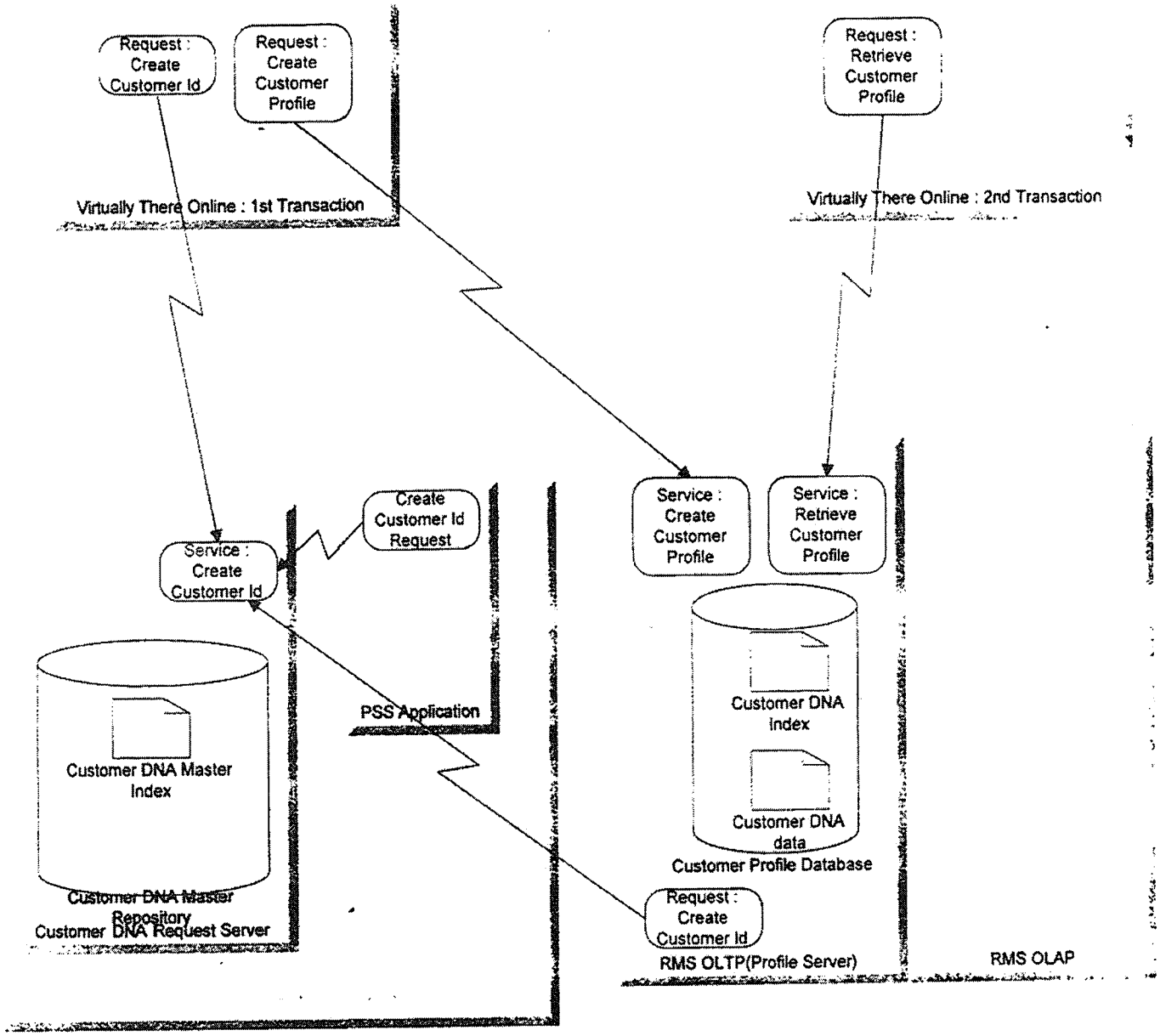
Additional Comments:

A request will be made to retrieve data from each system that carries customer information. (e.g. one for RMS, one for PNR data warehouse and one for Frequent Traveler Info).
It is the application that is responsible for accessing each respective database.

Customer DNA Services



Customer Services Usage Example



Considerations

- ***Input fields that will guarantee uniqueness*** - When there is a need to create a new Customer DNA number, we should query the DNA indices first to see if a number already exists for the customer. What if the only input fields will give a non-unique response, like name and address. We cannot really perform further verification because we may end up displaying data belonging to other travelers.

  Our solution to this is to adopt the rule of requiring the client application to send as part of its input query at least one field that will guarantee the uniqueness of the DNA number. For example, if a traveler has a customer profile already with a Customer DNA associated to it, and his newly created PNR is being ended, the ET package must send the FQTV number to see if a DNA exists for him. If there is no FQTV provided in the query, then instead of looking up to see if a DNA already exists, a new DNA will simply be created. This approach as already mentioned will require a nightly process to clean up dupes.

- ***Business constraints*** - Traveler name and phone number are mandatory PNR fields, but address is not. Are we going to be limited by existing constraints, or are we allowed to change the business rules?

- ***DNA number in STARS database*** - Data is uploaded from AATMS to the STARS database in PSS on a regular basis (either daily or weekly). Since AATMS will have customer DNA as an index, should we re-organize the STARS database to accommodate the DNA number? This will allow realtime retrieval of DNA number given a traveler's FQTV number. If we do this, the STARS application may have to be changed to allow STAR update via the addition of Customer DNA to an existing STAR record and to ensure that the data is downloaded to AATMS. Will there be a need for a STAR/Customer DNA index to facilitate retrieval of FQTV number/STAR record given a Customer DNA number?

- ***Security/ Viewership*** - Ticketing data in the VCR and T-REX databases are owned by the airlines and travel agencies, not the customers. Viewership issues will have to be resolved before access to the VCR and T-REX can be provided.

- ***DNA Generation*** - How do we generate Customer DNA Number – sequentially starting from 1, or using an algorithm (input to algorithm = name; address; phone number)?

What are the next steps?

- Validate detail design with Infrastructure.

- Flush out remaining elements of timeline.
- Communicate Customer DNA strategy to airline marketing organization.
- Integrate and align with other Enterprise Initiative projects.

Customer DNA
Phase I

Sabre

DETAIL DESIGN

for Business Development

v3.4

EXHIBIT 2

Prepared by
Sabre Inc.

Customer DNA

Detail Design

DOCUMENT REVISION INFORMATION

The following information is to be included with all versions of the document.

Project Name	CDNA	**Project Number**	MB321
Prepared by	Ceryl Medua, Brian Wong	**Date Prepared**	
Revised by	Ceryl Medua	**Date Revised**	
Revision Reason	Apply changes found during the review	**Revision Control No.**	1.0
Revised by	Ceryl Medua, Claudia Woodruff	**Date Revised**	
Revision Reason	Include Trillium Software	**Revision Control No.**	2.0
Revised by	Claudia Woodruff, Ceryl Medua, Mark Whitman	**Date Revised**	
Revision Reason	To define the CDNA objects	**Revision Control No.**	3.0
Revised by	Claudia Woodruff	**Date Revised**	
Revision Reason	Modifications resulting from Trillium Professional Services	**Revision Control No.**	3.4
Revised by		**Date Revised**	
Revision Reason		**Revision Control No.**	

Table of Contents

1 Introduction

1.1 Project Purpose/Scope

Customer DNA (CDNA) is an indexing system that will allow various applications within Sabre, Inc. (the organization) to access many of the other various databases within the organization, in order to get a consolidated view of the customer. The common link that will enable this consolidation of customer data is the CDNA number, which is a unique number assigned to each individual customer, and is allocated and stored within the CDNA. When a new customer record is created in any one of these internal systems, whether it is a PNR record, an RMS profile, a ticket record, or a Frequent Traveler Awards record, a CDNA number will be generated. The CDNA number, along with the prime key information from where the record originated (e.g. FQTV number, RMS ID, etc.) will be indexed into the CDNA system. The CDNA system will consist of a Cross Reference Table, which has the cross reference keys of subscribing customer databases, standardized tables for name, address and phone, email address, credit card information, and other identifying documentation such as passport number. For Phase 1, the cross reference keys that will be included in the Cross Reference Table are: RMS Index and PNR Index.

Design Criteria

2

2.1 CDNA Services – Process Overview

When customer data is added to a database within the one of the systems subscribing the CDNA, there is a need to make it possible for other systems to get to that customer data. The CDNA system will enable Sabre to map the distribution of the data pertaining to a customer by adding a cross reference item in CDNA. This is similar to the Dewey decimal card system catalog holding references to a book location.

When customer data is submitted for addition to the CDNA, a check must be done to determine if the customer already exists within CDNA. This is done by matching the new customer data to customer data already in CDNA. In order for this to occur with any degree of success, the customer name, address and phone number must be cleansed, and standardized before being matched with the cleansed and standardized data within CDNA. Much of the standardization and matching will be performed by calling modules of Trillium, a third party software package designed specifically for this purpose. Customer data is always cleansed and standardized before being added to the CDNA database.

The services provided for Phase I of Customer DNA are:

- Add a cross reference and, if this is a new customer, add new customer data.
- Delete a cross reference and, if there are no more references to the customer, delete customer data.
- Modify customer data – update of non-key information such as name, address, phone number, email address, etc.
- Retrieve the indices into the other databases for a specific customer.

There will be no modification to the cross reference (index) items since the data is considered to be prime keys in the subscribing system's database.

CDNA will only allow those systems who have subscribed to CDNA to perform any of the services listed. A system is known to be subscribed to CDNA when the systems External Interfacing Application (EIA) code exists in the CDNA EIA control table.

The EIA check is the only security check CDNA will perform. *It is the responsibility of all EIAs to perform their own security checks before giving customer data to any application. It is assumed by CDNA that whatever security is needed by the EIA's is in place, and outside the scope of the CDNA system.*

All of the services listed will only be available to EIAs via a Common Object Request Broker Architecture (CORBA).

2.1.1 Add a cross reference key

The process starts with a client request from an EIA such as BOA or RMS to add a cross reference key to CDNA system. The requesting EIA must supply their EIA code, customer data, and client system key. The customer data consists of attributes such as name and address. The client system key *(index item)* is the unique key by which a customer is known. These parameters are described in detail in the CDNA Interface Specification Document

The EIA code is checked first, and if it is not recognized by CDNA, an error will be returned, and no further processing will occur. The CDNA Cross-reference table is checked to see if the index item passed already exists.. The action taken in this case is described more fully in the detail section to follow. If the index does not exist, the customer data is then sent to Trillium for standardization and parsing. (See the Trillium Overview section of this document.) The submitted standardized customer data is then compared to the set of standardized customer data in the CDNA. Trillium will return a match if one exists. If there is no match, the submitted standardized customer data is added to the Customer tables(s) and a unique CDNA ID number is assigned to the customer. The index item is added to the Cross-reference table. If there is a match, the new index item is added to the Cross-reference table with the customer's existing CDNA ID number.

2.1.2 Delete a cross reference key

CDNA will delete the submitted index item from the Cross-reference table.

The customer data within CDNA for this specific customer may also be deleted. A check will be made to see if there are any additional index items in the cross-reference table for this specific customer. This is done by querying the table for the customer's CDNA ID number. The customer data will be deleted from CDNA database if no more indices to the customer exist.

2.1.3 Modify customer data

This service is to update the customer data only. As mentioned earlier, the client system keys may never be modified for a customer. The new data must be cleansed , standardized ,and parsed by Trillium before replacing the existing customer data and in order to recalculate the candidate key.

Using the client system key, the CDNA Cross-reference table will be queried to obtain the CDNA ID, in order to retrieve the customer data from the tables and then updated.

In a later phase, CDNA may publish the updated customer data, or publish a notice of a change in customer data in order for other subscribing systems to update their records. Alternately the other systems may simply request via the 'retrieve' service whenever they want to be sure their records are current.

2.1.4 Retrieve cross-reference keys

This occurs when an EIA requests all client system key(s) for a specific customer. All indexes for a specific customer are 'linked' together by the unique CDNA ID number.

Using the submitted index item, the CDNA ID number is determined for the customer. The Cross-reference table is queried using the CDNA ID number, resulting in a list of all indexes that are then returned to the submitting EIA.

2.2 Trillium Data Quality Software

The Trillium data quality software is the tool that will standardize and enhance the quality of data received from the external applications. The software will be used to match the input records against the records in the database. The four modules comprising the software are Converter, Parser, Geocoder, and Matcher. The Converter module is a general-purpose data transformation tool. The Parser module delivers standardized customer name and address data from unstructured or structured input. The Postal Geocoder utilizes the output of the Parser in order to validate and correct street level Postal Information. The Matcher is a generalized function used to determine relationships between transactions.

CDNA, the first Sabre system to use Trillium, will be customizing the four modules in order to match the highest percentage of customers as possible. Trillium will perform the majority of the processing for the CDNA system, however, to allow other EIAs to use its converting, parsing and geocoding routines, one instance of Trillium will exist and accessible as a CORBA service through the CDNA interface . At least at first, CDNA will 'own' and control the service. All customization will be done by CDNA. Other systems will not reconfigure the settings for any of the callable modules unless they go through the CDNA Change Management process.

2.2.1 Process Overview

Trillium has a batch processing mode, but CDNA phase I will use the callable modules in a real-time mode only. We will use the Match module in a 'reference matching' mode. This is where a record is compared to a group of records, referred to as a 'window'. Records in the window share commonality with the input record in some way. This strategy helps the matcher to be more efficient. A driver program, or class must be written for each of the four modules. In the batch processing, these driver programs are already supplied but for when calling the callable modules directly, we are responsible for writing our own.

Program Specifications

3

3.1 Index Class

3.1.1 addIndex

Description

Add index(es) to Customer DNA system.

Input Parameters

eiaCd

CustomerDataSeq (see cdna.idl for structure)

Return Values

void

Exceptions Thrown

ErrorB::CDNAException

Note: If at least one system key exists in the cross-reference table, the other index(es) will be added to the cross-reference table. The customer data will not be processed because there is no guarantee that it is more accurate or complete than the existing customer data.

Procedure

1.0 Validate EIA
2.0 For CustomerSeq.length
- 2.1 If customer.key already exists
  - 2.1.1 Write to error log, continue with next customer
- 2.2 If customer does not have minimum fields
  - 2.2.1 Write to error log, continue with next customer
- 2.3 convertOut = TB::convert(CustomerSeq[n])
- 2.4 parseIn = (selected fields from) convertOut
- 2.5 parseOut = TB::parse(parseIn)
- 2.6 geoCountry = parseOut.country
- 2.7 geocodeOut = TB::geocode(parseOut, geoCountry)
- 2.8 matchData = (selected fields from) geocodeOut
- 2.9 createWindow as follows:
  - 2.9.1 if matchData.fqtv

2.9.1.1 fqtvs = Pfqtv::queryFqtv(matchData.fqtv)
2.9.1.2 custs = Pcust::query(fqtv[n].cdnaId)
2.9.1.3 for each custs
2.9.1.3.1 windowItem[next].fields = cust[n].fields // assuming cust !null
2.9.1.3.2 addToWindow(windowItem)
2.9.2 if matchData.email
2.9.2.1 emails = Pemail::queryEmailAddr(matchData.email)
2.9.2.2 custs = Pcust::query(email[n].cdnaId)
2.9.2.2.1 windowItem[next].fields = cust[n].fields // assuming cust
2.9.2.2.2 addToWindow(windowItem)
2.9.3 Repeat as above with document, creditcard, phone
2.9.4 If matchData.candCd does not contain spaces (address available)
2.9.4.1 cands = Pcand::queryCandCd(matchData.candCd)
2.9.4.2 custs = Pcust::query(cand[n].cdnaId)
2.9.4.2.1 windowItem[next].fields = cust[n].fields
2.10 matchOut = match(matchData, window)
2.11 if no match
2.11.1 pcustomer = (selected fields from) matchOut
2.11.2 cdnaId = Pcustomer::add(pcustomer)
2.12 if more than one match
2.12.1 matched = select best one (based on criteria tbd)
2.12.2 cdnaId = matched.cdnaId
2.13 else (only one matched)
2.13.1 cdnaId = matched.cdnaId
2.14 todayDate = Time::today()
2.15 PXRef::add(cdnaId, clientsystemkey, todayDate)
3.0 Return

3.1.2 DeleteIndex()

Description

Delete customer index(es) to Customer DNA system. If was the last reference to a specific customer, then this method will invoke the deleteCustomer method.

Input Parameters

eiaCd

SystemKeySeq // See cdna.idl for structure

Return Values

void

Exceptions Thrown

ErrorB::CDNAException

Procedure

1.0 Validate Eia
2.0 For systmKeySeq.length
    2.1 xref = Pxref::queryKey(systemKey[n])
    2.2 cdnaId = xref.cdnaId
    2.3 Pxref::delete(xref)
    2.4 Xrefs = Pxref::queryKey(cdnaId)
    2.5 If no xrefs returned
        2.5.1 Pcust.delete(cdnaId)
3.0 Return successful

3.1.3 GetIndex()

Description

Retrieve the Indexes to a specific customer

Input Parameters

eiaCd

SystemKeyData

Return Values

SystemKeyDataSeq

Exceptions Thrown

ErrorB::CDNAException

Procedure

1.0 Validate Eia
2.0 xref = Pxref.queryKey(SystemKey)
3.0 xrefs = Pxref.queryKey(xref.cdnaId)
4.0 systemKeyDate[n] = xrefs[n].systemKeyData
5.0 return systemKeyData

3.2 ControlB Class

3.2.1 validEia

Description

Validates the external interfacing application (EIA) code

Input Parameters

EiaCd

Return Values

bool

Exceptions Thrown

ErrorB::CDNAException

Procedure

1.0 Initialized Indicator = FALSE
2.0 eia_var = eiaFactory->query(eiaCd)
3.0 If (!PS_is_nil())
    3.1 Indicator = TRUE
4.0 Return Indicator

3.2.2 ValidDocType()

Description

Validates the documentation type before allowing a document to be added.

Input Parameters

docType

Return Values

bool

Exceptions Thrown

ErrorB::CDNAException

Procedure

1.0 Initialized Indicator = FALSE
2.0 docType_var = docTypeFactory->query(docType)
3.0 If (!PS_is_nil())
    3.1 Indicator = TRUE
4.0 Return Indicator

3.3 CustomerB Class

3.3.1 modifyCustomer()

Description

Update customer data.

Input Parameters

eiaCd
SystemKeyData
Customer

Return Values

void

Exceptions Thrown

ErrorB::CDNAException

Procedure

1.0 Validate Eia
2.0 cdnaId = Pxref::queryKey(systemKeyData)
3.0 If no record, throw error exception
4.0 oldCust = Pcust::query(custId)
5.0 If address or phone fields exists in customer (fields to update include address or phone fields)
    5.1 ConvertIn = (fields from) customer
    5.2 convertOut = TB::convert(convertIn)
    5.3 parseIn = (selected fields from) convertOut
    5.4 parseOut = TB::parse(parseIn)
    5.5 geoCountry = parseOut.country
    5.6 geocodeOut = TB::geocode(parseOut, geoCountry)
    5.7 cdnaId =Pxref::queryKey(systemKeyData) // Find existing customer record
    5.8 oldCust.address = geocodeOut.address
    5.9 oldCust.phone = geocodeOut.phone (// do we add the phone or replace??)
6.0 Pemail::add(customer.emailAddr) // Do only if customer.emailAddr is not null , of course..
7.0 Pother::add(customer.other) // for creditcard, document etc.
8.0 createWindow as follows:
    8.1 if matchData.fqtv
        8.1.1 fqtvs = Pfqtv::queryFqtv(matchData.fqtv)
        8.1.2 custs = Pcust::query(fqtv[n].cdnaId)
        8.1.3 for each custs
            8.1.3.1 windowItem[next].fields = cust[n].fields // assuming cust !null
            8.1.3.2 addToWindow(windowItem)
    8.2 if matchData.email
        8.2.1 emails = Pemail::queryEmailAddr(matchData.email)
        8.2.2 custs = Pcust::query(email[n].cdnaId)
            8.2.2.1 windowItem[next].fields = cust[n].fields // assuming cust
            8.2.2.2 addToWindow(windowItem)
    8.3 Repeat as above with document, creditcard, phone

8.4 If matchData.candCd does not contain spaces (address available)
8.4.1 cands = Pcand::queryCandCd(matchData.candCd)
8.4.2 custs = Pcust::query(cand[n].cdnaId)
8.4.2.1 windowItem[next].fields = cust[n].fields
9.0 matchOut = match(matchData, window)
10.0 if no match
10.1 pcustomer = (selected fields from) matchOut
10.2 cdnaId = Pcustomer::add(pcustomer)
11.0 if more than one match
11.1 matched = select best one (based on criteria tbd)
11.2 cdnaId = matched.cdnaId
12.0 else (only one matched)
12.1 cdnaId = matched.cdnaId
13.0 todayDate = Time::today()
14.0 PXRef::add(cdnaId, clientsystemkey, todayDate)
15.0 Return successful

3.4 TB Class (Trillium Driver)

3.4.1 Tconvert()

Description

This is a general purpose driver to encapsulate formatting and manipulating the data and files necessary for the Trillium Converter callable module

Input Parameters

customerData

Return Values

convertOut

Exceptions Thrown

ErrorB::CDNAException

Procedure

This module will prepare the various buffers required by the Converter. The convertOut is the output from the converter callable module. See Trillium documentation for details specific to the callable modules.

3.4.2 Tparse()

Description

This is a general purpose driver to encapsulate formatting and manipulating the data and files necessary for the Trillium Parser callable module

Input Parameters

parseIn

Return Values

parseOut

Exceptions Thrown

ErrorB::CDNAException

Procedure

This module will prepare the various buffers required by the Parser. The parseOut is the output from the converter callable module. See Trillium documentation for details specific to the callable modules.

3.4.3 Tgeocode()

Description

This is a general purpose driver to encapsulate formatting and manipulating the data and files necessary for the Trillium Geocoder(s) callable module

Input ParametersData

ParseOut, country

Return Values

convertOut

Exceptions Thrown

ErrorB::CDNAException

Procedure

This module will prepare the various buffers required by the geocoder. The geocodeOut is the output from the geocoder callable module

There are requirements specific per country. See Trillium documentation. See Trillium documentation for details specific to the callable modules.

DRAFT

Customer DNA Phase I

Sabre

INTERFACE SPECIFICATIONS

for Business Development

version 1.0

EXHIBIT 3

Prepared by
Sabre Inc.

Date

Customer DNA

Interface Specifications

DOCUMENT REVISION INFORMATION

The following information is to be included with all versions of the document.

Project Name	Customer DNA	**Project Number**	MB321
Prepared by	Ceryl Medua, Steven Brooks	**Date Prepared**	
Revised by	Ceryl Medua, Claudia Woodruff	**Date Revised**	
Revision Reason	Applied changes from CDNA IDL	**Revision Control No.**	1.0
Revised by	Ceryl Medua, Claudia Woodruff	**Date Revised**	
Revision Reason	Include Trillium in CDNA IDL	**Revision Control No.**	1.1
Revised by		**Date Revised**	
Revision Reason		**Revision Control No.**	
Revised by		**Date Revised**	
Revision Reason		**Revision Control No.**	
Revised by		**Date Revised**	
Revision Reason		**Revision Control No.**	
Revised by		**Date Revised**	
Revision Reason		**Revision Control No.**	

Customer DNA

Interface Specifications

DOCUMENT APPROVAL

This signature page is to indicate approval from Sabre sponsor and Client sponsor for the attached Interface Specifications for the Customer DNA project. All parties have reviewed the attached document and agree with its contents.

Sabre Project Manager
Rick Sharp, Customer DNA Project Manager

Date

Client Project Manager
Greg Gibson, Product Manager, CRM, TM&D

Date

Sabre Sponsor
Karen Carter, Director, Customer /-Services, Customer Content

Date

Client Sponsor
Barbara Norton-Boyle, Director, CRM, TM&D

Date

Other
Name, Title

Date

Table of Contents

1 Introduction

1.1 Interface Specifications Overview

The Interface Specifications document identifies the protocols, data flows, processes and procedures between the Customer DNA System and the external interfacing application (EIA). It is intended for use by the customer and development teams in the validation of the business requirements and in the management of the software development process, and by the developers and analysts in the development or modification of the system.

This document addresses the interface specifications between Customer DNA System and the external systems such as Relationship Management System (RMS) and Buyer Offload Analysis Data Distributor (BOA DD).

For Phase I, the cross-reference keys of external systems that will be included in the Customer DNA system are RMS Id and PNR locator. The services to be provided are adding a cross reference key, deleting a cross reference key, updating non-key information, and retrieving cross reference keys.

System Overview

2

2.1 EIA Overview

The Relationship Management System (RMS) and the Buyer Offload Analysis Data Distributor (BOA DD) are the External Interfacing Application (EIA) for Phase I.

The Relationship Management System (RMS) focuses on the development of an enterprise customer-centric repository for Sabre. RMS will contain complete, accurate views of Sabre customers, with ties to their associated data. RMS is envisioned to be the centralized source of customer data for the organization. It will integrate and store customer-centric data that is currently fragmented across multiple sources within the organization. Customer data from each system will be integrated to provide whole, unified views of Sabre customers.

The Buyer Offload Analysis Data Distributor (BOA DD) focuses on developing a production quality infrastructure. The hub consists of a relational database on a scalable UNIX platform that contains traveler data updated in near real time. BOA proved that it was possible to integrate TPF and UNIX platforms in order to successfully transmit TPF data from PSS to a UNIX-based Informix ODS database.

2.2 Customer DNA Overview

Customer DNA is an indexing system that will allow Sabre applications to access the various customer databases in Sabre and give a consolidated view of the customer. The common link that will enable this consolidation of customer data is the Customer DNA number, which is a unique number assigned to each individual customer. When a customer record is created in Sabre, whether it is a PNR record, a ticket record, a profile record, or a Frequent Traveler Awards record, a Customer DNA number will be generated. The Customer DNA number, along with the prime key information from where the record originated (e.g. FQTV number, RMS ID, etc.) will be indexed into the Customer DNA system. The Customer DNA system will comprise of a table, which has the cross reference keys to access the information from all the customer databases, standard tables for name, address and phone, where the Customer DNA number is generated, and match items table for non-prime key data.

The Trillium data quality software is the tool that will standardize and enhance the quality of data received from the external interfacing applications for Customer DNA. The software, through its matching functionality, will eliminate duplicate occurrences of a customer in the Customer DNA database. The four modules comprising the software are Converter, Parser, Geocoder, and Matcher. Customer DNA system will customize the four modules in order to

match the highest percentage of customers as possible. To allow other External Interfacing Applications (EIAs) to use it specifically for data cleansing, Trillium will be provided as a CORBA service.

2.3 Other Source Systems

Phase I of Customer DNA will provide data access only to the Relationship Management System (RMS) and the Buyer Offload Analysis Data Distributor (BOA DD) applications. No other external systems are designed to interface with Customer DNA during this initial implementation.

3 Assumptions and Constraints

3.1 General Assumptions and Constraints

1. Phase I of Customer DNA will only interface with Relationship Management System (RMS) and the Buyer Offload Analysis Data Distributor (BOA DD) applications. No end-user interface will be provided beyond what will be developed and implemented for RMS and BOA DD. The communication link with the end-user will be established and maintained through the external interfacing application (EIA).
2. No analytical features will be developed or implemented to facilitate decision support functions of external systems.
3. Implementation of Customer DNA will be within a client/server CORBA environment.
4. A limited bundling of services will be provided.
5. All access to Customer DNA database will be through the Customer DNA services that will be published as they are finalized.
6. Fully implemented services, i.e., to other Sabre applications, will be provided in future phases of Customer DNA.

3.2 Data Volume

According to *Customer DNA Volumetrics for PSS* document, an estimated 31 million new PNR End Transactions (ET) are generated each month in PSS. The PNR is sent to BOA DD at End Transaction (ET) time. Phase I implementation of Customer DNA will impact PNR EM entry, which a subset of ET. The volume that EM will bring is assumed to be lower because the entry is new and the actual measurement of its use is not available yet.

RMS has estimated that 100 million Declared Traveler Profiles will exist after three years of operation.

3.3 Data Automation Requirements

Customer DNA will be a service in Sabre's Client-Server Architecture as defined by the Travel Distribution Framework (TDF) Group. Data will be transferred between the client's service via CORBA IDL (Interface Definition Language).

3.4 Audit Requirements

No audit requirements are identified in the Customer DNA Position Paper.

4 EIA Data Interface Specifications

4.1 Adding/Deleting CDNA Cross Reference Keys

4.1.1 Interface Description

This interface handles the addition and deletion of cross reference keys in Customer DNA system when a request occurs from the external interfacing application such as RMS and BOA DD. A new RMS profile or a new PNR triggers an add request to this interface. The cross reference keys of the external interfacing application is sent across, together with the other data that the receiving application (i.e. Customer DNA) needs. A delete request to this interface occurs when the external interfacing application sends a request to remove the supplied cross reference key from Customer DNA system. This happens in BOA DD when a PNR is purged. After Customer DNA has processed the request, a successful or error response is sent back, whatever the case may be.

4.1.2 Assumptions and Constraints

A limit to the number of customers that can be added during one call may be set depending on the level of communication traffic.

An unsuccessful response will be returned if a customer to be added does not have the required minimum data.

4.1.3 Type of Interface

The interface is in IDL (Interface Definition Language) format and will be implemented in CORBA.

4.1.4 Input (EIA to Customer DNA)

The table below describes the input from external interfacing application (i.e. RMS and BOA DD) to Customer DNA.

Element Name	Description
SYSTEM KEY	a cross reference key of a system supplied by an external interfacing application that is unique for each customer. This may be an RMS Id (for RMS system) or a PNR locator plus the vendor code, create date and name association number (for BOA DD).
NAME	includes first name, last name and middle name (if present) of the traveler. This is required from BOA DD and RMS for adding a cross reference key.

Element Name	Description
ADDRESS	includes street, city, state, zip code, country, and address type.
PHONE NUMBER	may be home, work or cell phone associated to the traveler.
FQTV NUMBER	includes frequent traveler number and its vendor code
EMAIL ADDRESS	an address associated with a person(s) which is used when sending electronic messages on computers attached to local or global networks.
DOCUMENT INFO	a form of identification of the traveler such as passport, driver's license, etc.
CREDIT CARD INFO	credit card number of the traveler and does not include credit card vendor.

4.1.5 Process Detail

A request from external interfacing application to add a cross reference to Customer DNA assigns a Customer DNA number to the cross reference data that will be added to the Customer DNA database.

A request from external interfacing application to delete a cross reference from Customer DNA removes the cross reference key of the external application and its associated data from the Customer DNA database.

The *Customer DNA Detail Design* presents more details on these processes.

4.1.6 Output (Customer DNA to EIA)

There is no expected output. Success or failure of a service call is determined through an exception thrown by CORBA.

4.2 Updating CDNA Information

4.2.1 Interface Description

This interface handles the modification of information other than cross reference keys in Customer DNA system when a request occurs from the external interfacing application such as RMS and BOA DD. A change in customer information from external application such as new address or phone number triggers a modification request to this interface. The new information sent to the receiving application (i.e. Customer DNA) includes the cross reference key of the external interfacing application to identify the item to be changed. After Customer DNA has processed the request, a successful or error response is sent back, whatever the case may be.

4.2.2 Assumptions and Constraints

There will be no modification of a cross reference key since the data is considered to be prime key in the database.

4.2.3 Type of Interface

The interface is in IDL (Interface Definition Language) format and will be implemented as CORBA services.

4.2.4 Input (EIA to Customer DNA)

The table below describes the input from external interfacing application (i.e. RMS and BOA DD) to Customer DNA.

Element Name	Description
SYSTEM KEY	a cross reference key of a system supplied by an external interfacing application that is unique for each customer. This may be an RMS Id (for RMS system) or a PNR locator plus the vendor code, create date and name association number (for BOA system).
NAME	includes first name, last name and middle name (if present) of the traveler. This is required from BOA DD and RMS for adding a cross reference key.
ADDRESS	includes street, city, state, zip code, country, and address type.
PHONE NUMBER	includes street, city, state, zip code, country, and address type.
FQTV NUMBER	includes frequent traveler number and its vendor code
EMAIL ADDRESS	an address associated with a person(s) which is used when sending electronic messages on computers attached to local or global networks.
DOCUMENT INFO	a form of identification of the traveler such as passport, driver's license, etc.
CREDIT CARD INFO	credit card number of the traveler and does not include credit card vendor.

4.2.5 Process Detail

A request from external interfacing application to modify non-key information associated to a cross reference key in Customer DNA adds the non-key information to the Customer DNA database.

The *Customer DNA Detail Design* presents more details on these process.

4.2.6 Output (Customer DNA to EIA)

There is no expected output. Success or failure of a service call is determined through an exception thrown by CORBA.

4.3 Retrieving CDNA Cross Reference Keys

4.3.1 Interface Description

This interface handles the request from the external interfacing application, such as RMS and BOA DD, for a list of cross reference keys of another external system. If RMS, for example, wants a list of the PNR locators associated to a particular customer who has a profile in RMS, this is the interface to use. The Customer DNA system, when it receives the request, sends back to the external interfacing application the list of cross reference items requested.

4.3.2 Assumptions and Constraints

The limit to the number of cross reference items that may be returned to the requesting application will be determined and set during construction.

An unsuccessful response will be returned if cross reference items do not exist in the Customer DNA database.

4.3.3 Type of Interface

The interface is in IDL (Interface Definition Language) format and will be implemented as CORBA services.

4.3.4 Input (EIA to Customer DNA)

The table below describes the input from external interfacing application (i.e. RMS and BOA DD) to Customer DNA.

Element Name	Description
SYSTEM KEY	a cross reference key of a system supplied by an external interfacing application that is unique for each customer. This may be an RMS Id (for RMS system) or a PNR locator plus the vendor code, create date and name association number (for BOA system).

4.3.5 Process Detail

An external interfacing application requesting a list of cross reference keys (of a particular customer or traveler) to another external system populates the data structure defined in the

IDL with the key of the requesting application and the type of list of cross reference keys needed. A call to the interface function is invoked to send the request across, then waits for the response.

The *Customer DNA Detail Design* presents more details on these process.

4.3.6 Output (Customer DNA to EIA)

Customer DNA returns a list of cross reference keys to the external application requested if the request is processed successfully. Failure of a service call is determined through an exception thrown by CORBA.

5 Application Programming Interface

5.1 Customer DNA APIs

The following APIs are drawn from the IDL for Customer DNA system. The complete listing of the IDL including the definition of the data structures can be found in the Appendix.

5.1.1 addCDNAXref()

Name

addCDNAXref

Adds cross reference item(s) in Customer DNA database.

Syntax

```
addCDNAXref (string *pEiaCd,
             CDNACustomerSeq *pCDNACustomers);
```

Arguments

Name	Description
pEiaCd	pointer to external interfacing application code that sends the request
pCDNACustomers	pointer to customer data to be added to the Customer DNA repository such as system key, name, address, phone, email, etc.

Returns

Name	Description
ET_SUCCESS	indicates that the process was completed successfully.
CDNAException	indicates that an error occurred.

5.1.2 deleteCDNAXref()

Name

deleteCDNAXref

Deletes cross reference item(s) in Customer DNA database.

Syntax

```
deleteCDNAXref (string *pEiaCd,
                SystemKeyDataSeq *pSystemKeys);
```

Arguments

Name	Description
pEiaCd	pointer to external interfacing application code that sends the request
pSystemKeys	pointer to a collection of system keys to be deleted from the Customer DNA repository such as PNR locator and create date, RMS id, etc.

Returns

Name	Description
ET_SUCCESS	indicates that the process was completed successfully.
CDNAException	indicates that an error occurred.

5.1.3 modifyCustomerData()

Name

modifyCustomerData

Modifies non-key information based on the supplied cross reference item in Customer DNA database.

Syntax

```
modifyCustomerData (string *pEiaCd,
                    CDNACustomerSeq *pCDNACustomers);
```

Arguments

Name	Description
pEiaCd	pointer to external interfacing application code that sends the request
pCDNACustomers	pointer to customer data to be modified such as name, address, phone, email, etc.

Returns

Name	Description
ET_SUCCESS	indicates that the process was completed successfully.
CDNAException	indicates that an error occurred.

5.1.4 getCDNAXref()

Name

getCDNAXref

Retrieves cross reference item(s) in the Customer DNA repository.

Syntax

```
getCDNAXref (string *pEiaCd,
             SystemKeyData *pSystemKey);
```

Arguments

Name	Description
pEiaCd	pointer to external interfacing application code that sends the request
pSystemKey	pointer to the search argument system key

Returns

Name	Description
SystemKeyDataSeq	pointer to a list of system keys retrieved. Indicates successful process.
CDNAException	Indicates that an error occurred.

5.2 Trillium APIs

5.2.1 standardize()

Name

standardize

Calls Trillium software to cleanse and standardize the customer data.

Syntax

```
standardize (string *pEiaCd,
             TCustomerSeq *pTCustomers);
```

Arguments

Name	Description
pEiaCd	pointer to external interfacing application code that sends the request
pTCustomers	pointer to a collection of customer data to be standardized by Trillium software

Returns

Name	Description
ET_SUCCESS	indicates that the process was completed successfully.
CDNAException	indicates that an error occurred.

6 Other Interface Requirements

6.1 Security Requirements

1. Only an authorized EIA may access Customer DNA. A Customer DNA-internal control table will maintain the valid EIAs.

2. Customer DNA APIs will be the sole method for EIAs to access the Customer DNA database.

3. Refer to *Customer DNA Security Design* document for additional security requirements.

6.2 Standard Data

Standard data is derived from data received from the clients that has been "standardized" (i.e. cleansed and transformed) using a data quality tool, such as Trillium. Currently, this data is stored in the Customer DNA system for operational purposes. No requirements exist to provide this data to other systems. In the future, however, this data may be required by other applications, such as Sabre's CRM (Customer Relationship Management) system. This need may derive updates and additions to the existing APIs.

7 Appendices

7.1 References and Related Documentation

- *Customer DNA Position Paper*
- *Customer DNA Project Definition Document*
- *Customer DNA Architecture Document*
- *Customer DNA Security Design Document*
- *Customer DNA Detail Design Document*
- *Customer DNA Logical Model*
- *Customer DNA Physical Model (not available at this time)*

7.2 Customer DNA IDL

```
//============================================================================
// File:   cdna.idl (Customer DNA IDL)
// Author: Claudia Woodruff & Ceryl Medua
// Date:
// Description: Interface specification for the CDNA services
// Modified:              - to include Trillium
//============================================================================

module CDNA
{
    struct Address
    {
        string stAddr1;
        string stAddr2;
        string geoline1;  // For city, state, zip, country etc
        string geoline2;

        // Null flags
        boolean stAddr1NULL;
        boolean stAddr2NULL;
        boolean geoline1NULL;
        boolean geoline2NULL;
    };

    struct Document
    {
        string cryCode;
        string type;
        string number;
    };

    struct SystemKeyData
    {
        string client;
```

```
        string clientSystemKey;
    };
    typedef sequence <SystemKeyData> SystemKeyDataSeq;

    struct CDNACustomer
    {
        SystemKeyData   key;
        string          name;
        Address         address;
        string          fqtvNr;
        string          phone;
        string          creditCard;
        string          emailAddr;
        Document        document;

        // Null flags
        // NOTE: no nameNULL - name is mandatory;
        // At lease one of the following must be false (i.e. present)
        boolean addressNULL;
        boolean fqtvNrNULL;
        boolean phoneNULL;
        boolean creditCardNULL;
        boolean emailAddrNULL;
        boolean documentNull;
    };
    typedef sequence <CDNACustomer> CDNACustomerSeq;

    struct TCustomer  // To be standardized by Trillium only
    {
       string       name;
       Address      address;
       // Null flags
       // NOTE: no null flags - name and address is mandatory
    };
    typedef sequence <TCustomer> TCustomerSeq;

    typedef sequence <string> standardizedCustSeq;

// ---------------
// Interface Begin
// ---------------

    exception CDNAException
    {
    unsigned short  code;
             string dateTime;
             string name;
             string desc;
    };

    interface CDNASession;
    interface CDNASessionFactory;

    interface CDNASessionFactory
    {
       CDNASession   create(in string name);
    };

    interface CDNASession
    {
       // Add a cross-reference item
       //(with optional additional system keys)
       void addCDNAXref(in string eiaCd,
```

```
                        in CDNACustomerSeq customers)
                        raises (CDNAException);

        // Deleting cross reference item
        void deleteCDNAXref (in string eiaCd,
                             in SystemKeyDataSeq keys)
                             raises (CDNAException);

        // Customer data maintenance
        void modifyCustomerData(in string eiaCd,
                                in CDNACustomerSeq customers)
                                raises (CDNAException);

        // Retrieve cross-reference items for a customer
        SystemKeyDataSeq getCDNAXref(in string eiaCd,
                                     in SystemKeyData key)
                                     raises (CDNAException);

        // Convert, Parse, and Geocode customer name and address
        standardizedCustSeq standardize(in string eiaCd,
                                        in TCustomerSeq tCustomers)
                                        raises (CDNAException);
    };
};
```

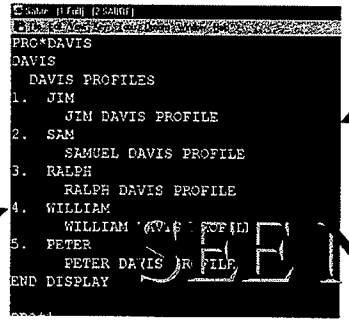
8 Project Glossary

8.1 Glossary of Terms

Refer to Customer DNA Glossary. <\\$nds\.project1_proj1c.projects.com.dev\CLHPMO\Customer DNA\CDNA Glossary.doc>

CRM at the Travel Agent desktop

Customer Contacts
Travel Agency



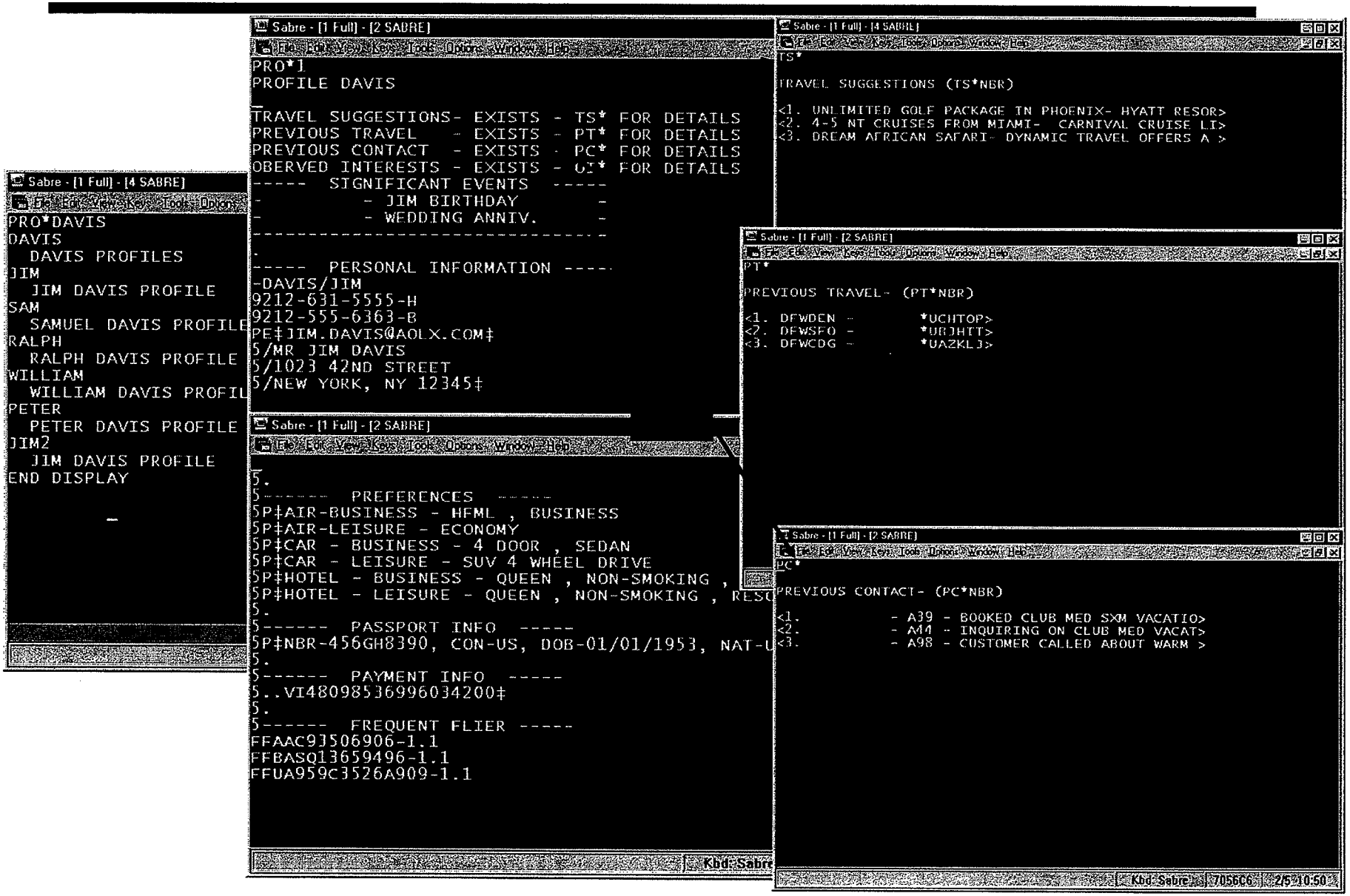
Pop-Up Customer
Profile Screen

Travel Agent Invokes CRM
Tools via Sabre for
Windows

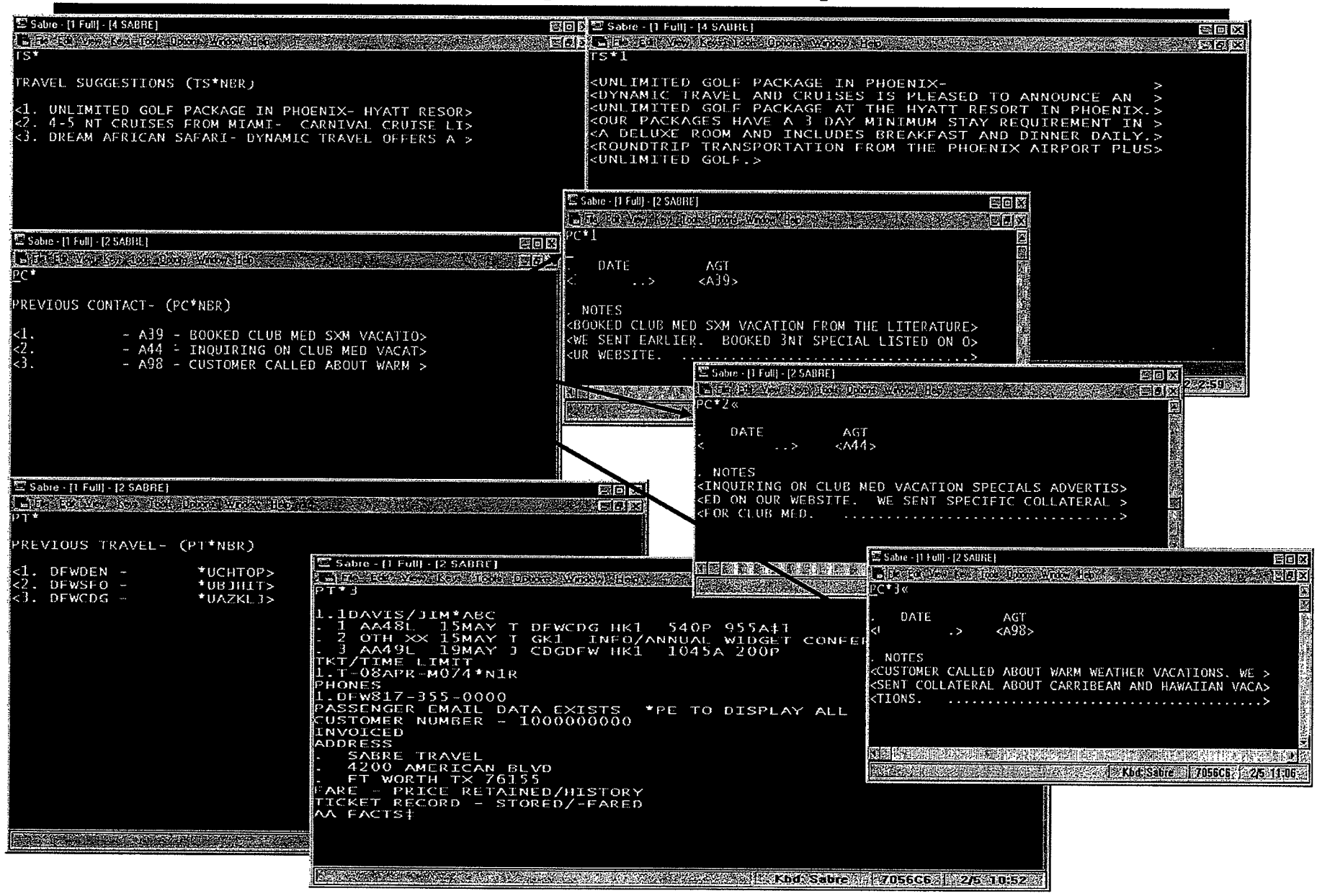
SEE NEXT SLIDE...

EXHIBIT 4

CRM at the Travel Agent desktop



CRM at the Travel Agent desktop



WHAT IS THE INNOVATION?

- Sabre Labs proposes to demonstrate how Sabre can provide operational (as opposed to analytical) CRM functionality to the travel agent desktop, specifically the 'green screen' Sabre for Windows desktops. The envisioned functionality will provide Sabre green screen mask-oriented displays of traveler profile data from RMS and PNR data from BOA (for now until the PNR Data Warehouse is operational). These displays will be triggered by a series of new 'profile display / edit' commands and the existing 'PNR display' commands. Included will be the ability to display and edit RMS data, display prior trips / PNR's, create a written log of agent-to-customer interactions, and review prior customer contact logs. Subsequent phases of this project may address: a) Integration of CTI technology such that the above commands can be triggered from a customer phone call in addition to agent-on-demand; b) "Fan-out" / integration of the above functionality to all other Sabre / agency points of customer contact such as eVoya and WebRes; c) A complete, multi-touchpoint interaction center for the travel agency that includes call center integration (ACD, PBX, IVR, etc), automated email interaction and management, and webchat.; d) Integration of a travel recommendation engine which includes but is not limited to Sabre Labs collaborative filtering work as well as Dynamic Packaging and Itinerary Genie.

WHAT IS THE POTENTIAL VALUE TO THE BUSINESS?

- Opportunity for Sabre to provide basic CRM services to agencies sooner, rather than later (with the eVoya migration plan) – especially the wide-spread installed base of Sabre-for-Windows users.

EVALUATION CRITERIA:

Key Evaluation Areas	Findings
Are current Sabre defined viewership / security controls with regard to customer profile STARS and agency PNRs still enforceable in the proposed design / prototype?	As of the timeframe when this prototype was being constructed, the RMS team had not yet designed in the necessary defined viewership / security controls because their first RMS implementation in was solely for the needs of VTO. The RMS team does intend to have these controls in place by the time any sort of multi-Sabre-customer information is present (such as multiple agencies or multiple airlines).
Do the BOA and RMS designs support the envisioned functionality, particularly the envisioned fetch, display, edit and create functionality?	Yes, as expected and required, RMS does support create, read, update and delete of profile records. And BOA supports read only as expected and required.
Can this be delivered effectively via green screen Sabre?	Yes, as evidenced by the pre-production demonstration completed in
Internal reactions to the prototype from the Sabre Agency Solutions and Sabre CRM departments	TMD CRM department is pleased with the mock-up screens and prototype. They are now reviewing findings from 54 in-depth interviews with agencies to determine which features to prioritize for green screen vs. eVoya

EXHIBIT 5

External / usability test reactions from travel agencies	Preliminary findings are that travel agencies like the profile management and prior trip functionality best. They also like the contact management and suggestion functionality but don't fully understand how to efficiently integrate it into their workflow. Those that have evolved to mature practices with ClientBase also like the functionality but are skeptical as to Sabre's intentions with traveler profile data.

FINAL DELIVERABLE(S):

- A prototype demonstrating RMS and BOA 'screen pops' on a Sabre for Windows green screen Sabre session.

IMPLEMENTATION FACTORS:

- Must integrate with the Sabre CRM ASP platform. Must ensure the same or better viewership controls that agencies have become accustomed to with profile STARS and PNR's.

BUSINESS UNIT APPLICABILITY: **SABRE TMD TRAVEL AGENCY SOLUTIONS; SABRE TMD CRM**

BUSINESS UNIT INVOLVEMENT: **SABRE TMD CRM (JOEL BAILEY, MICHAEL ASKEW, HUNTLEY MCNAB, BARBARA NORTON BOYE); SABRE TMD TRAVEL AGENCY SOLUTIONS (ELLEN KESZLER)**

THIRD PARTY VENDORS: **BORLAND / INPRISE (FOR VISIBROKER CORBA SOFTWARE)**

CONTACT INFORMATION: **EDDIE CASH (963-6427)**

RECENT ACTIVITIES:

- Completed initial pre-production prototype capable of fetching and displaying basic traveler profile data from RMS along with past PNR's depicting prior travel history from BOA. Have halted further development work pending funding review with TMD CRM group.
- 54 travel agency feedback sessions were completed. Findings indicated that travel agencies like the profile management and prior trip functionality best. They also like the contact management and suggestion functionality but don't fully understand how to efficiently integrate it into their workflow.
- Prototype has been demonstrated and transferred to business owner Barbara Norton Boyle (TMD CRM) who i using the prototype and feedback session to define and prioritize requirements for the CTO development grou headed by Cathy Harshman.
- Participated in BOA/RMS/CDNA capacity planning session on 5/18 conducted by the CTO office.

RELATED INFORMATION:

- The TMD CRM group completed a business requirements document (BRD) dated 4/30 that describes the RM Phase II business requirements. These requirements were based largely on concepts demonstrated in the CF at TA Desktop prototype, including traveler profile management and trip history retrieval and display. An RMS Phase II project kick-off is scheduled for
- Future Sabre Labs CRM work will focus on Travel Suggester and CRM at the Airline / Airport Agent Desktop.

MILESTONES:

Task	Status	Plan	Revised Plan	Actual
Define prototype requirements and mockup	Complete	12/8		12/8
Assemble prototype development team	Complete	12/19		12/19
Understand BOA and RMS databases and access methods	Complete	1/12	1/26	1/26
Prototype design	Complete	1/19	1/26	1/26
Install and configure Corba ORB for accessing RMS	Complete	1/19		2/9
Complete code for RMS access using the RMS Corba API	Complete	1/26		2/9
Complete SQL and related code for accessing BO	Complete	1/26		1/26
Complete TPF code for PNR display from BOA	Complete	2/2	2/16	3/9
Complete first demonstration	Complete	2/16	3/29	3/29
Complete TPF code for profile and contact history display from Twister	Complete	2/2	2/16	3/9
Complete RMS access configuration for profile and contact history display from RMS	Halted	2/2	3/2	3/9
Complete TPF code for 'Now Move'	Halted	3/16		3/9
Complete TPF code for create, update and delete of RMS profiles	Halted	3/16		3/9
Complete prototype construction and transfer to TMD CRM	Complete	3/30		3/30
Complete technology transfer to TMD business unit and CTO development group	In progress	6/29		

INTELLECTUAL PROPERTY:

- None